ERB-947

NRCC No. 20925

National Research Council Canada
Conseil national de recherches Canada

Division of Electrical Engineering
Division de génie électrique

ALGORITHMS FOR PIECEWISE LINEAR CHEBYSHEV APPROXIMATION OF PLANE CURVES

N.N. ABDELMALEK

DECEMBER 1982

40

Canada

Copyright 1982 by
National Research Council of Canada

Permission is granted to quote short excerpts and to reproduce figures and tables from this report, provided that the source of such material is fully acknowledged.

Copyright 1982 par
Conseil national de recherches du Canada

Il est permis de citer de courts extraits et de reproduire des données ou tableaux du présent rapport, à condition d'en identifier clairement la source.

Additional copies are available free of charge from:

Editorial Office, Room 301
Division of Electrical Engineering
National Research Council of Canada
Ottawa, Ontario, Canada
K1A 0R8

Des exemplaires supplémentaires gratuits peuvent être obtenus à l'adresse suivante:

Bureau des publications, Pièce 301
Division de génie électrique
Conseil national de recherches du Canada
Ottawa (Ontario) Canada
K1A 0R8

ALGORITHMS FOR PIECEWISE LINEAR CHEBYSHEV APPROXIMATION OF PLANE CURVES

N.N. Abdelmalek

ERB-947

NRCC No. 20925

DECEMBER 1982

TABLE OF CONTENTS

	Page
ABSTRACT	v
RÉSUMÉ	vi
1. INTRODUCTION	1
2. APPLICATIONS OF THE ALGORITHMS	2
3. CHARACTERISTICS OF THE APPROXIMATION	2
4. THE DISCRETE LINEAR CHEBYSHEV APPROXIMATION PROBLEM	3
5. THE TWO PROGRAMS	4
6. NUMERICAL RESULTS AND COMMENTS	4
ACKNOWLEDGEMENTS	7
REFERENCES	7
APPENDIX	9
Driver DR1	10
Subroutine PC1	13
Driver DR2	17
Subroutine PC2	21
Subroutine LINFIN	29

LIST OF FIGURES

1	Disconnected Chebyshev piecewise approximation with vertical parabolas with pre-assigned tolerance = 1	5
2	Connected Chebyshev piecewise approximation with vertical parabolas with pre-assigned tolerance = 1	5
3	Disconnected Chebyshev piecewise approximation with vertical parabolas. Near balanced norm solution with number of segments = 4	6

ABSTRACT

Two computer subroutines and their subsidiary routine, written in FORTRAN IV, for solving the piecewise linear Chebyshev approximation problem of plane curves are given. The two subroutines are written for two respective algorithms described in a recent paper by the author (Int. J. Syst. Sci. 14: 1983). The first algorithm is for the case when the Chebyshev error norm in any segment is not to exceed a pre-assigned value. The second algorithm is for the case when the number of segments is given and a balanced Chebyshev error norm solution is required. The given curve is first digitized and either algorithm is then applied to the discrete points. Both algorithms use parametric linear programming techniques, which result in speed of computation. They also take advantage of the equioscillation property of the Chebyshev approximation and thus avoid unnecessary computation. Hence the algorithms are faster than other known methods. Also no constraints on the approximating functions are required. Given as well are two Drivers for the given subroutines. An numerical example is presented.

RÉSUMÉ

Ce rapport décrit deux sous-routines d'ordinateur et leur routine subsidiaire, écrites en FORTRAN IV, pour la résolution du problème de l'approximation restreinte linéaire discrète de Chebyshev des courbes planes. Les deux sous-routines sont écrites respectivement pour les deux algorithmes décrits dans un article récent du même auteur (Int. J. Syst. Sci. 14: 1983). Le premier algorithme s'applique au cas ou le vecteur résiduel dans la norme de Chebyshev relatif à un segment quelconque ne doit pas dépasser une valeur prédéterminée. Le deuxième algorithme s'applique au cas où le nombre de segments est donné et où la solution doit donner des vecteurs résiduels de Chebyshev équilibrés. La courbe est d'abord numérisée et l'un ou l'autre des algorithmes appliqué aux points ainsi relevés. Les deux algorithmes sont basés sur des techniques de programmation paramétrique linéaire donnant une grande vitesse de calcul; ils utilisent la propriété d'équioscillation de l'approximation de Chebyshev pour éliminer des calculs inutiles. Ces caractéristiques leur confèrent une rapidité supérieure à celle des autres méthodes connues. Ces algorithmes présentent en outre l'avantage de ne pas imposer de contraintes aux fonctions d'approximation. Sont précisés deux programmes de gestion pour les deux sous-routines et un exemple numérique est donné.

FORTRAN PROGRAMS FOR PIECEWISE LINEAR CHEBYSHEV APPROXIMATION OF PLANE CURVES

Nabih N. Abdelmalek

1. INTRODUCTION

The problem of piecewise approximation in the Chebyshev norm has received considerable attention for some time. Lawson (1) derived characteristic properties of the segmented Chebyshev approximation problem and proposed an iterative algorithm for finding such an approximation. Phillips (2) described an algorithm for obtaining approximations to a given function by means of straight line segments with a pre-assigned tolerance. Later, Pavlidis (3) described an algorithm for waveform segmentation through functional approximation. Again Pavlidis and Maika (4) proposed a procedure based on functional iteration for solving the same problem. In a recent paper, Kioustelidis (4) discusses the existence of segmented approximation with free knots for a given continuous function.

A deficiency in all piecewise Chebyshev approximation algorithms is that, at each iteration, the approximation for each segment is recalculated from scratch. That is, no information from a previous iteration is reserved and used in a present iteration. See, for example, the arithmetic operations count in Pavlidis and Maika (ref. 4, p. 67). Also none of the algorithms takes advantage of the principal property of the Chebyshev approximation, namely the equioscillation property (6, 7). This property helps in avoiding unnecessary computation. Again, the majority of published methods consider only the special cases of piecewise approximation by constants or by straight lines.

In this work, two computer subroutines and their subsidiary routine, written in FORTRAN IV, for solving the piecewise linear Chebyshev approximation problem of plane curves are given. The two subroutines are written for two respective algorithms described in a recent paper by the author (8). The first algorithm is for the case when the Chebyshev error norm in any segment does not exceed a pre-assigned value. The second algorithm is for the case when the number of segments is given and a (balanced) Chebyshev error norm solution is required.

Given also are two Drivers for the given subroutines.

The problem is solved by first digitizing the given curve. Then either algorithm is applied to the discrete points. Both algorithms apply

parametric linear programming techniques to an existing discrete linear Chebyshev approximation algorithm developed by Abdelmalek (9, 10). That is, information from a previous iteration is reserved and used in a present iteration. Thus less computation is needed at each iteration. Both algorithms also take advantage of the equioscillation property in the Chebyshev approximation. Hence unnecessary computation is avoided. The algorithms are thus faster than other known methods. Finally no conditions are imposed on the approximating functions. They need not be constants or straight lines as is the case in the majority of the published methods.

2. APPLICATIONS OF THE ALGORITHMS

Piecewise linear continuous approximation of plane curves has several applications. In the field of numerical analysis, it is used in the reduction of nonlinear problems to approximately equivalent linear problems. It is also used in the approximate solutions to linear differential equations with time varying parameters (see Solodov (11)).

Examples from electrical engineering are in analog computers (12) and the design of real-time digital filters.

In the fields of image processing and pattern recognition, it is used in feature extraction, noise filtering and data compression (13) (see, for example, Pavlidis (14)). Hence it is extensively used in robotic vision systems.

It is also applied in speech recognition and numerous other fields.

3. CHARACTERISTICS OF THE APPROXIMATION

For convenience the following notation is used. Let us be given the plane curve $y = f(x)$ defined on the interval $[\alpha, \beta]$. Let this curve be digitized at the K points $\{x_i, f(x_i)\}$, $i = 1,\ldots,K$, where $x_1 = \alpha$ and $x_K = \beta$. Let n be the number of segments (pieces) in the approximation. Finally let $\{z_j\}$, $j = 1,\ldots,n$, be the n Chebyshev norms of the n segments.

Assuming f(x) is continuous and satisfies Lipschitz condition on the given interval, Lawson (1) showed that the norm z_j, $1 \leq j \leq n$, has the following properties:

(a) z_j is a continuous function of the end points of segment j

(b) z_j is nonincreasing in the left end point of segment j
(c) z_j is nondecreasing in the right end point of segment j

Lawson also showed that if a piecewise Chebyshev approximation is calculated for the curve f(x) (not for the digitized points of the curve), and if

[1] $$z_1 = z_2 = \ldots = z_m$$

then the solution is optimal. Also an optimal solution with this property always exists. Such a solution is known as a balanced error norm solution.

However, in the case of the approximation of the discrete points resulting from digitizing the given curve, a balanced error norm solution may not exist. One may in this case attempt to minimize the variance of the error norms $\{z_j\}$ (see Pavlidis (3), p. 690).

4. THE DISCRETE LINEAR CHEBYSHEV APPROXIMATION PROBLEM

Consider any segment j, say $1 < j < n$, of the given curve f(x). Let this segment consist of N digitized points with coordinates $\{x_i, f(x_i)\}$ i=1,...,N. Let these N points be approximated by the function

[2] $$P(a, x) = \sum_{j=1}^{M} a_j \phi_j(x)$$

which minimizes the Chebyshev norm z_j (or briefly z), of the residuals $r(x_i)$, where

[3a] $$z = \max_i |r(x_i)|$$

and

[3b] $$r(x_i) = P(a, x_i) - f(x_i), \quad i = 1,\ldots,N$$

In [2] $\{\phi_j(x_i)\}$, $j = 1,\ldots,M$, $M < N$, is a set of given approximating functions and 'a' is an M-vector to be calculated. For example, if the approximating function is a straight line, $P(a, x) = a_1 + a_2x$, where a_1 and a_2 are the elements of vector 'a' .

This problem reduces to the problem of obtaining the Chebyshev solution of the overdetermined system of linear equations (6)

[4] $$Ca = f$$

C is a given real N × M matrix of rank $k \leq M < N$ given by $C = (\phi_j(x_i))$, $i = 1,\ldots,N$ and $j = 1,\ldots,M$ and f is an N-vector where $f = f(x_i)$. The Chebyshev solution to system [4] is the real M vector a^* which minimizes the Chebyshev norm

[5] $$z = \max_i |r_i(a)|, \quad i=1,\ldots,N$$

where r_i is the ith residual in [4] and is given by [3b] or by

[6] $$r_i = c_{i1}a_1 + \ldots + c_{iM}a_M - f_i$$

This problem has been solved using linear programming techniques by Abdelmalek (9) and an efficient FORTRAN routine has also been written (10).

5. THE TWO PROGRAMS

Two computer programs entitled PC1 AND PC2 are written in FORTRAN IV for the present two algorithms, respectively. They are tested on a number of problems in single precision computation on the VAX 11/780 computer. Each of these programs uses the subroutine LINFIN as a subsidiary program. The subroutine LINFIN is a slight modification to the subroutine LINF (10).

Program PC1 has the option of calculating connected or disconnected piecewise Chebyshev approximations, while PC2, because of its nature, can only calculate disconnected piecewise Chebyshev approximations. In the connected piecewise approximation, the end point of segment j say, is the starting point of segment (j + 1). In the disconnected piecewise approximation, the adjacent point to the right of the end point of segment j is the starting point of segment (j + 1).

6. NUMERICAL RESULTS AND COMMENTS

The numerical results of one example are presented here. The given curve for this example is that of figure 5 in Tomek (15). The curve is digitized with equal x-intervals into 28 points. The data points are fitted with vertical parabolas, each is of the form $y = a_1 + a_2x + a_3x^2$.

For program PC1, a parameter ρ is given, where the Chebyshev error norm in each segment is $\leq \rho$. Figures 1 and 2 are the results of program PC1 where the approximation is disconnected and connected, respectively, and $\rho = 1$ for both cases.

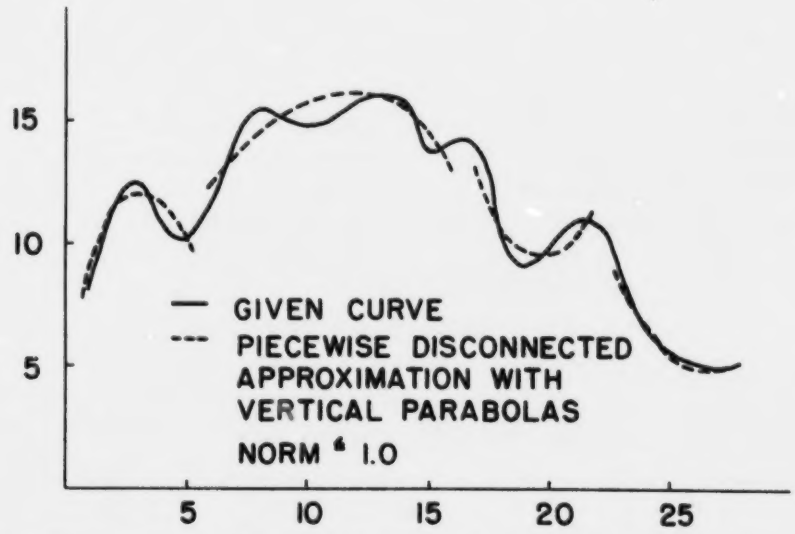

Figure 1. Disconnected Chebyshev piecewise approximation with vertical parabolas with pre-assigned tolerance = 1.

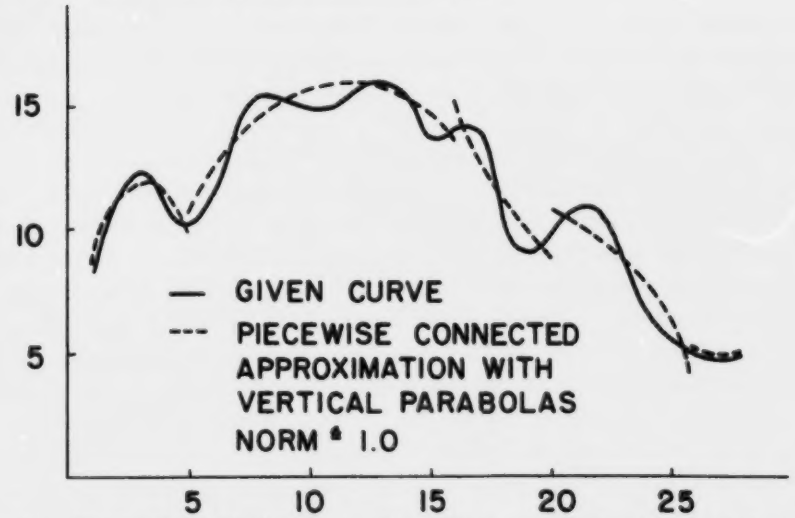

Figure 2. Connected Chebyshev piecewise approximation with vertical parabolas with pre-assigned tolerance = 1.

For program PC2, the number of segments n as well as the initial starting point for each segment is given. Figure 3 shows the results of program PC2 for n = 4. The initial starting points are calculated by having an equal number of points for each segment.

The Chebyshev norms for Figs. 1, 2 and 3 are, respectively, $\{0.3944, 0.9696, 0.6917, 0.1594\}$, $\{0.3944, 0.9696, 1.0, 0.8876, 0.0\}$ and $\{0.3944, 0.9696, 0.4667, 0.2188\}$. We notice that program PC2 did not

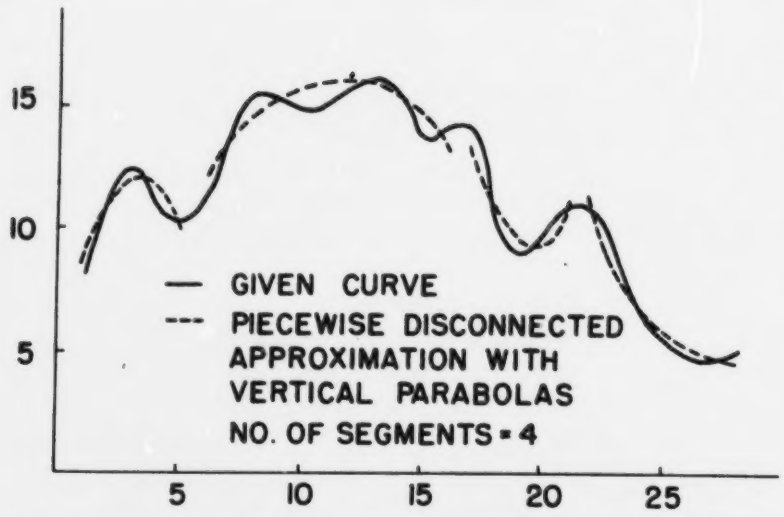

Figure 3. Disconnected Chebyshev piecewise approximation with vertical parabolas. Near balanced norm solution with number of segments = 4.

produce a balanced norm solution (equal Chebyshev norms for all the segments). However, it gave an improved solution to that of PC1. Compare Figs. 3 and 1. See also the comments at the end of Section 3.

The results of the test examples we used are again obtained by modifying programs PC1 and PC2 in such a way that in each iteration the piecewise approximation for each segment is calculated from scratch. The CPU times for the modified programs are larger than the corresponding CPU times for the unmodified programs. This depends on the particular example and the manner the curve is initially segmented. A ratio up to 2.5 times is obtained for either PC1 or PC2. This ratio for the results for Figs. 1, 2 and 3 is 1.4, 2.4 and 1.9, respectively.

We conclude that the present programs are faster than other known methods due to two factors. The first one is that we used the program LINF (10) which is written specifically for Chebyshev approximation. Other authors use less efficient programs, mainly all purpose linear programming routines for calculating the Chebyshev approximation (see, for example, Pavlidis and Maika (4)). The second factor is the techniques used in this paper in applying the program LINF. Finally, no conditions are imposed on the approximation functions [2]. They need not be constants or straight lines as in the majority of published methods.

ACKNOWLEDGEMENTS

I wish to sincerely thank the staff of the Editorial Office of the division of Electrical Engineering of the NRC for their assistance in the publication of this and my previous reports. The help of Denis Saumure in drawing the illustrations is very much appreciated.

REFERENCES

1. Lawson, C.L. Characteristic properties of the segmented rational minimax approximation problem. Numer. Math. 6: 293-301; 1964.

2. Phillips, G.M. Algorithms for piecewise straight line approximations. Comput. J. 11: 211-212; 1968.

3. Pavlidis, T. Waveform segmentation through functional approximation. IEEE Trans. Comput., C-22: 689-697; 1973.

4. Pavlidis, T., and Maika, A.P. Uniform piecewise polynomial approximation with variable joints. J. Approx. Theory. 12: 61-69; 1974.

5. Kioustelidis, J.B. Optimal segmented approximations. Computing, 24: 1-8; 1980.

6. Cheney, E.W. Introduction to Approximation Theory. McGraw-Hill, New York, NY. 1966.

7. Osborne, M.R., and Watson, G.A. On the best linear Chebyshev approximation. Comput. J. 10: 172-177; 1967.

8. Abdelmalek, N.N. Piecewise linear Chebyshev approximation of plane curves. Int. J. Syst. Sci., 14: 1983.

9. Abdelmalek, N.N. Chebyshev solution of overdetermined systems of linear equations, BIT, 15: 117-129; 1975.

10. Abdelmalek, N.N. A computer program for the Chebyshev solution of overdetermined systems of linear equations. Int. J. Numer. Methods Eng. 10: 1197-1202; 1976.

11. Solodov, A.V. Linear Automatic Control Systems with Varying Parameters. Blackie, London, England. 1966.

12. Korn, G.A., and Korn, T.M. Electronic Analog and Hybrid Computers. McGraw-Hill, New York, NY. 1964. Chapt. 6.

13. Davison, L.D. Data compression using straight line interpolation. IEEE Trans. Inf. Theory, IT-14: 390-394; 1968.

14. Pavlidis, T. The use of algorithms of piecewise approximations for picture processing applications. ACM Trans. Math. Software, 2: 305-321; 1976.

15. Tomek, I. Two algorithms for piecewise-linear continuous approximation of p functions of one variable. IEEE Trans. Comput. C-23: 445-448; 1974.

APPENDIX

DRIVERS AND SUBROUTINE LISTINGS

```
C     THIS PROGRAM IS A DRIVER FOR THE SUBROUTINE PC1  WHICH
C CALCULATES THE PIECEWISE LINEAR CHEBYSHEV APPROXIMATION OF
C A GIVEN DATA POINT SET (X,Y) RESULTING FROM THE  DISCRETI-
C ZATION OF A PLANE CURVE Y=F(X).
C     FROM THE APPROXIMATING FUNCTIONS WE FORM  THE  OVERDE-
C TERMINED SYSTEM OF LINEAR EQUATIONS C*A=F.  (SEE TEXT).
C    THE APPROXIMATION BY SUBROUTINE PC1 IS  SUCH  THAT  THE
C CHEBYSHEV  NORM  FOR  ANY  SEGMENT IS NOT TO EXCEED A PRE-
C ASSIGNED TOLERANCE 'ENORM'.
C    THE CALCULATION IS IN SINGLE PRECISION.
C    THE MAXIMUM NUMBER OF POINTS (NUMBER OF ROWS IN  MATRIX
C  C) IN THIS PROGRAM IS 250 AND THE MAXIMUM NUMBER OF TERMS
C IN THE APPROXIMATING FUNCTION IS 25.  THESE CAN BE  EASILY
C CHANGED BY CHANGING THE PARAMETERS NN AND MM BELOW.
C    THIS DRIVER CONTAINS 2 EXAMPLES.
C
      REAL*4 TIME,ORIGIN
      DIMENSION CT(26,250),F(250),R(250),A(26),BINV(26,26),
     1BV(26),IBOUND(250),ICBAS(26),IRBAS(26)
      DIMENSION CTN(26,250),FN(250)
      DIMENSION IRANKP(50),RP(250),AP(50,26),ZP(50),
     1IXL(50),V(26)
      DIMENSION FB(44),FC(28)
C
      DATA FB/10.0,12.0,13.3,13.1,13.5,15.0,17.8,17.5,
     116.4,15.3,16.0,16.2,14.5,11.5,9.9,9.7,9.3,7.0,
     24.5,3.4,3.8,4.1,3.0,1.9,2.1,4.8,6.0,6.2,6.1,6.5,
     311.0,13.3,13.2,13.1,14.0,16.5,17.8,16.6,15.2,
     415.3,16.1,15.8,13.0,10.0/
C
      DATA FC/8.0,11.0,12.4,11.2,10.1,11.0,13.8,15.4,
     115.1,15.0,15.1,15.8,16.0,15.6,13.5,14.3,14.1,
     210.6,8.9,9.9,10.5,10.8,9.0,6.8,5.8,5.0,4.8,4.9/
C
  101 FORMAT(10H1EXAMPLE  ,I3,8H      N=,I3,8H      M=,I3)
  102 FORMAT(40H0F(J)                                    )
  103 FORMAT(40H0TRANSPOSE OF MATRIX C                   )
  104 FORMAT(17H0EXECUTION TIME =,F10.5,13HSECONDS       )
  105 FORMAT(30H0  IXL(J):J=1,NPIECE          )
  106 FORMAT(25H0    IRANKP                )
  107 FORMAT(25H0    I,  AP                )
  108 FORMAT(25H0    ZP                    )
  109 FORMAT(25H0    RP                    )
  110 FORMAT(1H ,10F10.4)
  111 FORMAT(1H ,10I5)
  112 FORMAT(1H ,I5,2X,7F10.4)
      NN=250
```

```
      MM=26
      KK=50
      EPS=1.E-4
      IEXMPL=0
    1 IEXMPL=IEXMPL+1
      GO TO (7,9,100), IEXMPL
C EXAMPLE 1.
    7 N=44
      M=3
      ENORM=0.8
      DO 8 J=1,N
        E=FLOAT(J)
        EE=E*E
        EEE=E*EE
        F(J)=FB(J)
        CT(1,J)=1.0
        CT(2,J)=E
        CT(3,J)=EE
    8   CT(4,J)=EEE
        GO TO 80
C EXAMPLE 2.
    9 N=28
      M=3
      ENORM=1.0
      DO 10 J=1,N
        E=FLOAT(J)
        EE=E*E
        EEE=E*EE
        F(J)=FC(J)
        CT(1,J)=1.0
        CT(2,J)=E
        CT(3,J)=EE
   10   CT(4,J)=EEE
   80 CONTINUE
      KONECT=0
      KONECT=1
      M1=M+1
      WRITE(3,101)IEXMPL,N,M
      WRITE(3,102)
      WRITE(3,110)(F(J),J=1,N)
      WRITE(3,103)
      DO 81 I=1,M1
   81   WRITE(3,110)(CT(I,J),J=1,N)
      CALL SETCLK(ORIGIN)
      CALL PC1(M,N,MM,NN,KK,CT,F,EPS,ICBAS,IRBAS,BINV,BV,
     1IBOUND,IRANK,ITER,R,A,Z,IND,ENORM,CTN,FN,
     2IRANKP,RP,AP,ZP,IXL,V,NPIECE,KONECT)
      CALL RDCLK(TIME,ORIGIN)
      WRITE(3,104)TIME
      WRITE(3,105)
      WRITE(3,111)(IXL(J),J=1,NPIECE)
```

```
      WRITE(3,106)
      WRITE(3,111)(IRANKP(I),I=1,NPIECE)
      WRITE(3,107)
      DO 82 I=1,NPIECE
   82   WRITE(3,112)I,(AP(I,J),J=1,M1)
      WRITE(3,108)
      WRITE(3,110)(ZP(I),I=1,NPIECE)
      WRITE(3,109)
      WRITE(3,110)(RP(J),J=1,N)
      GO TO 1
  100 STOP
      END
```

```
      SUBROUTINE PC1(M,N,MM,NN,KK,CT,F,EPS,ICBAS,IRBAS,
     1BINV,BV,IBOUND,IRANK,ITER,R,A,Z,IND,ENORM,CTN,FN,
     2IRANKP,RP,AP,ZP,IXL,V,NPIECE,KONECT)
C
C     GIVEN IS A SET OF POINTS (X,Y) RESULTING FROM DIGI-
C TIZING A GIVEN PLANE CURVE OF THE FORM Y-F(X). FROM THE
C APPROXIMATING FUNCTIONS WE FORM THE OVERDETERMINED
C SYSTEM OF LINEAR EQUATIONS C*A-F. (SEE TEXT).
C     THIS PROGRAM IS A SUBROUTINE FOR CALCULATING A
C LINEAR PIECEWISE CHEBYSHEV APPROXIMATION TO THE DISCRETE
C POINT SET (X,Y). THE APPROXIMATION IS SUCH THAT THE
C CHEBYSHEV NORM FOR ANY SEGMENT IS NOT TO EXCEED A PRE-
C ASSIGNED TOLERANCE 'ENORM'. THE NUMBER OF SEGMENTS OR
C PIECES IS NOT KNOWN.
C     THIS SUBROUTINE USES PARAMETRIC LINEAR PROGRAMMING
C TECHNIQUES TO AN EXISTING DISCRETE LINEAR CHEBYSHEV
C APPROXIMATION SUBROUTINE. IT ALSO MAKES USE OF THE EQUI-
C OSCILLATION PROPERTY OF THE CHEBYSHEV APPROXIMATION.
C    THE SUBROUTINE IS WRITTEN IN SINGLE PRECISION. IT MAY
C BE CONVERTED TO DOUBLE PRECISION BY MERELY REPLACING THE
C STATEMENT
C     DOUBLE PRECISION S,SA
C BY THE STATEMENT
C     IMPLICIT REAL*8(A-H,O-Z)
C     THE INPUT DATA TO THE SUBROUTINE.
C M       THE NUMBER OF TERMS IN THE APPROXIMATING FUNCTION.
C N       THE TOTAL NUMBER OF POINTS TO BE PIECEWISE APP-
C         ROXIMATED.
C MM      AN INTEGER GREATER THAN OR EQUAL TO (M+1).
C NN      AN INTEGER GREATER THAN OR EQUAL TO N.
C KK      AN INTEGER LARGE ENOUGH SUCH THAT THE TOTAL NUMBER
C         OF SEGMENTS IS TO BE LE.KK.
C CT      A MATRIX OF DIMENSIONS MM BY NN. ITS FIRST M ROWS
C         AND FIRST N COLUMNS CONTAIN THE TRANSPOSE OF
C         MATRIX C OF THE SYSTEM C*A-F. MATRIX CT DOES NOT
C         CHANGE IN THE COMPUTATION.
C F       AN NN VECTOR. ITS FIRST N ELEMENTS CONTAIN THE
C         R.H.S. OF THE SYSTEM C*A-F. VECTOR F DOES NOT
C         CHANGE IN THE COMPUTATION.
C EPS     A SPECIFIED TOLERANCE SUCH THAT A CALCULATED NUMBER
C         X IS CONSIDERED ZERO IF !X!<EPS. FOR THE VAX 11/
C         780 THE ROUND-OFF LEVEL IS ABOUT 1.E-6 AND 1.E-16
C         FOR SINGLE AND DOUBLE PRECISION RESPECTIVELY. FOR
C         THIS MACHINE, EPS IS USUALLY TAKEN 1.E-4 AND 1.E-11
C         FOR SINGLE AND DOUBLE PRECISION CALCULATION RESPCT.
C ENORM   A REAL NUMBER; A PRE-ASSIGNED VALUE, SUCH THAT
C         THE CHEBYSHEV NORM FOR ANY SEGMENT IS .LE.ENORM.
C KONECT AN INTEGER SUCH THAT IF KONECT.EQ.0, THE SUBROUTINE
C         CALCULATES DISCONNECTED CHEBYSHEV PIECEWISE APPROX-
C         MATION. IF KONECT.NE.0, THE SUBROUTINE CALCULATES
C         CONNECTED CHEBYSHEV PIECEWISE APPROXIMATION.
```

```
C      THE OUTPUT OF THE SUBROUTINE.
C NPIECE AN INTEGER WHICH IS THE NUMBER OF SEGMENTS (PIECES)
C        OF THE APPROXIMATION;  NPIECE.LE.KK.
C CTN    AN MM BY NN MATRIX.  ON EXIT CTN CONTAINS THE FINAL
C        VALUES OF THE SIMPLEX TABLEAU.
C FN     AN NN VECTOR.  ON EXIT FN HAS THE SAME ELEMENTS  OF
C        VECTOR F ABOVE BUT WITH PLUS OR MINUS SIGNS.
C IRANKP AN KK VECTOR.  ITS FIRST  NPIECE  ELEMENTS  CONTAIN
C        THE  RANK  VALUES  OF  MATRICES  C IN THE EQUATIONS
C        C*A-F FOR THE NPIECE SEGMENTS.  IN CASE OF  PERFECT
C        FIT FOR SEGMENT J SAY, WE GET IRANKP(J)=0.
C RP     AN NN VECTOR.  ITS FIRST  N  ELEMENTS  CONTAIN  THE
C        RESIDUAL  VALUES  OF  THE  N  GIVEN  POINTS  OF THE
C        GIVEN CURVE.
C AP     AN KK BY MM MATRIX.  THE FIRST M  ELEMENTS  OF  ITS
C        FIRST  ROW  CONTAIN  THE  ELEMENTS  OF THE SOLUTION
C        VECTOR A* FOR  THE  FIRST  SEGMENT.   THE  FIRST  M
C        ELEMENTS  OF  ITS  SECOND  ROW CONTAIN THE SOLUTION
C        VECTOR A* FOR THE SECOND SEGMENT AND SO ON.  IF THE
C        FIRST  M ELEMENTS OF ROW J SAY, ARE ALL ZEROS, THIS
C        INDICATES THAT VECTOR A* OF ROW J IS NOT CALCULATED
C        BECAUSE THE NUMBER OF POINTS OF SEGMENT J IS <(M+1)
C        AND THERE IS A PERFECT FIT FOR SEGMENT J.
C ZP     AN KK VECTOR.  ITS FIRST  NPIECE  ELEMENTS  CONTAIN
C        THE  NPIECE  OPTIMUM  VALUES OF THE CHEBYSHEV NORMS
C        FOR THE NPIECE SEGMENTS.  IF ZP(J)=0.0, THIS  INDI-
C        CATES THAT THERE IS A PERFECT FIT FOR SEGMENT J.
C IXL    A KK-VECTOR.  ITS FIRST NPIECE ELEMENTS CONTAIN THE
C        INDEXES   OF  THE  FIRST  ELEMENTS  OF  THE  NPIECE
C        SEGMENTS.  FOR EXAMPLE IF IXL=(1,5,12,22,  ...   ),
C        THEN  THE FIRST SEGMENT CONTAINS POINTS 1 TO 4, THE
C        SECOND SEGMENT CONTAINS THE POINTS  5  TO  11,  THE
C        THIRD SEGMENT CONTAINS THE POINTS 12 TO 21 ...  .
C     THE MEANING OF THE OTHER PARAMETERS.
C ICBAS,IRBAS,BINV,BV,IBOUND,IRANK,ITER,R,A,Z,IND:       ARE
C        PARAMETRS  TO BE PASSED TO THE AUXILIARY SUBROUTINE
C        'LINFIN' AND THEY HAVE THE SAME MEANINGS AS IN  THE
C        SUBROUTINE  LINF.   SUBROUTINE LINFIN IS A SLIGHTLY
C        MODIFIED VERSION OF SUBROUTINE  LINF  FOR  DISCRETE
C        LINEAR CHEBYSHEV APPROXIMATION.  (SEE THE TEXT).
C V      AN MM WORKING VECTOR.
C
      DOUBLE PRECISION S,SA
      DIMENSION CT(MM,NN),F(NN),R(NN),A(MM),BINV(MM,MM),
     1BV(MM),IBOUND(NN),ICBAS(MM),IRBAS(MM)
      DIMENSION CTN(MM,NN),FN(NN),IRANKP(KK),RP(NN),
     1AP(KK,MM),ZP(KK),IXL(KK),V(MM)
      NPIECE=1
      IS=1
      M1=M+1
      KASE2=0
```

```
 1 KASE1=0
   IE=IS+M
   ZP(NPIECE)=0.0
   IRANKP(NPIECE)=0
   DO 2 J=IS,IE
 2   RP(J)=0.0
   DO 3 I=1,M1
 3   AP(NPIECE,I)=0.0
 4 IF(IE.GT.N) IE=N
   IXL(NPIECE)=IS
   NU=IE-IS+1
   IF(NU.LT.M1)GO TO 14
   IF(KASE1.NE.0) GO TO 6
   DO 5 J=IS,IE
     FN(J)=F(J)
     DO 5 I=1,M1
 5     CTN(I,J)=CT(I,J)
 6 CALL LINFIN(M,N,IS,IE,MM,NN,CTN,FN,EPS,ICBAS,IRBAS,
  *BINV,BV,IBOUND,IRANK,ITER,R,A,Z,IND,KASE1,KASE2)
   IF(Z.GT.ENORM) GO TO 13
   IRANKP(NPIECE)=IRANK
   ZP(NPIECE)=Z
   DO 7 J=1,M1
 7   AP(NPIECE,J)=A(J)
   DO 8 J=IS,IE
 8   RP(J)=R(J)
   KASE1=-1
   IF(IRANK.LT.M) KASE1=0
 9 J=IE+1
   IF(J.GT.N) GO TO 14
   IBOUND(J)=1
   FN(J)=F(J)
   CT(M1,J)=1.0
   DO 10 I=1,M1
     K=IRBAS(I)
10   V(I)=CT(K,J)
   DO 12 I=1,M1
     S=0.0
     SA=-F(J)-Z
     DO 11 K=1,M1
       S=S+BINV(I,K)*V(K)
11     SA=SA+A(K)*CT(K,J)
     CTN(I,J)=S
12   ZJ=SA
   ZP(J)=ZL
   IE=J
   IF(ZJ.LT.0.0) ZJ=-ZJ
   IF(ZJ.GT.Z) GO TO 4
   GO TO 9
13 IS=IE-1
   IF(KONECT.EQ.0) IS=IE
```

```
      NPIECE=NPIECE+1
      GO TO 1
   14 CONTINUE
      RETURN
      END
```

```
C     THIS PROGRAM IS A DRIVER FOR THE SUBROUTINE PC2  WHICH
C CALCULATES THE PIECEWISE LINEAR CHEBYSHEV APPROXIMATION OF
C A GIVEN DATA POINT SET (X,Y) RESULTING FROM THE  DISCRETI-
C ZATION OF A PLANE CURVE Y=F(X).
C     FROM THE APPROXIMATING FUNCTIONS WE FORM  THE  OVERDE-
C TERMINED SYSTEM OF LINEAR EQUATIONS C*A=F.  (SEE TEXT).
C     GIVEN:1)AN INTEGER NUMBER 'NPIECE' WHICH IS THE NUMBER
C OF  SEGMENTS  IN  THE APPROXIMATION AND 2) INITIAL INDEXES
C OF THE FIRST POINT OF EACH OF THE  NPIECE  SEGMENTS.  THE
C SUBROUTINE  PC2  CALCULATES  A  LINEAR PIECEWISE CHEBYSHEV
C APPROXIMATION TO THE DISCRETE POINT SET (X,Y), SUCH THAT A
C BALANCED  CHEBYSHEV ERROR NORM IS ATTEMPTED FOR THE NPIECE
C SEGMENTS (PIECES).
C    THE MAXIMUM NUMBER OF POINTS (NUMBER OF ROWS IN  MATRIX
C  C) IN THIS PROGRAM IS 250 AND THE MAXIMUM NUMBER OF TERMS
C IN THE APPROXIMATING FUNCTION IS 25.  THESE CAN BE  EASILY
C CHANGED BY CHANGING THE PARAMETERS NN AND MM BELOW.
C    THE CALCULATION IS IN SINGLE PRECISION.
C    THIS DRIVER CONTAINS 4 EXAMPLES.
C
      REAL*4 TIME,ORIGIN
      DIMENSION CT(26,250),F(250),R(250),A(26),BINV(26,26),
     1BV(26),IBOUND(250),ICBAS(26),IRBAS(26)
      DIMENSION FA(25),FB(44),FC(28)
      DIMENSION CTN(26,250),CTP(26,250),FN(250),FP(250)
      DIMENSION AL(26),BINVL(26,26),IRBASL(26),ICBASL(26),
     1RL(250),IBOUNL(250),BVL(26)
      DIMENSION AR(26),BINVR(26,26),IRBASR(26),ICBASR(26),
     1RR(250),IBOUNR(250),BVR(26)
      DIMENSION IRBASP(50,26),ICBASP(50,26),BINVP(50,26,26),
     1IBOUNP(250),IRANKP(50),RP(250),AP(50,26),ZP(50),
     2IXL(50),ICHANG(50),V(26)
C
      DATA FA/.0872673,.0872794,.0873029,.0873315,.0873576,
     1.3491184,.3498802,.3513824,.3532572,.3550109,
     2.6111334,.6150641,.6230824,.6336395,.6441493,
     3.8733883,.8841621,.9071868,.9400757,.9766021,
     41.135895,1.157550,1.206257,1.283258,1.384432/
C
      DATA FB/10.0,12.0,13.3,13.1,13.5,15.0,17.8,17.5,
     116.4,15.3,16.0,16.2,14.5,11.5,9.9,9.7,9.3,7.0,4.5,
     23.4,3.8,4.1,3.0,1.9,2.1,4.8,6.0,6.2,6.1,6.5,
     311.0,13.3,13.2,13.1,14.0,16.5,17.8,16.6,15.2,15.3,
     416.1,15.8,13.0,10.0/
C
      DATA FC/8.0,11.0,12.4,11.2,10.1,11.0,13.8,15.4,15.1,
     115.0,15.1,15.8,16.0,15.6,13.5,14.3,14.1,10.6,8.9,
     29.9,10.5,10.8,9.0,6.8,5.8,5.0,4.8,4.9/
C
  101 FORMAT(10H1EXAMPLE  ,I3,8H      N=,I3,8H      M=,I3)
  102 FORMAT(40H0F(J)                                    )
```

```
  103 FORMAT(40H0TRANSPOSE OF MATRIX C                    )
  104 FORMAT(17H0EXECUTION TIME =,F10.5,13HSECONDS        )
  112 FORMAT(30H0  IXL(J):J=1,NPIECE          ,20I5)
  113 FORMAT(30H0    NPIECE   =               ,I5)
  114 FORMAT(25H0    I,  ICBASP           ,20I5)
  115 FORMAT(25H0    I,  IRBASP           ,20I5)
  116 FORMAT(4H0 I=,I5,6H,AP  =,10E12.5)
  117 FORMAT(25H0    BINVP                )
  118 FORMAT(25H0    ZP                   ,10F9.5)
  119 FORMAT(25H0    RP                   ,10F9.5)
  120 FORMAT(1H ,10E12.5)
  122 FORMAT(25H0DRIVER DR2PRINT          )
  130 FORMAT(25H0    IRANKP               ,20I5)
C
      NN=250
      MM=26
      KK=50
      EPS=1.E-4
      IEXMPL=0
    1 IEXMPL=IEXMPL+1
      GO TO (2,5,7,9,100), IEXMPL
C EXAMPLE 1.
    2 M=3
      N=25
      DO 3 J=1,N
    3   F(J)=FA(J)
      DO 4 J=1,5
        D=0.15*(J-3)
        DD=D*D
        DDD=D*DD
        DO 4 I=1,5
          E=0.15*(I-3)
          EE=E*E
          EEE=E*EE
          K=5*(J-1)+I
          CT(1,K)=1.
          CT(2,K)=D
          CT(3,K)=E
          CT(4,K)=DD
          CT(5,K)=EE
          CT(6,K)=E*D
          CT(7,K)=DDD
          CT(8,K)=EEE
          CT(9,K)=DD*E
    4     CT(10,K)=EE*D
      GO TO 80
C EXAMPLE 2
    5 M=2
      N=25
      DX=1.0/(N-1)
      DO 6 J=1,N
```

```
        E=DX*FLOAT(J-1)
        EE=E*E
        F(J)=EXP(E)
        CT(1,J)=1.0
        CT(2,J)=E
    6   CT(3,J)=EE
      GO TO 80
C EXAMPLE 3.
    7 N=44
      M=3
      DO 8 J=1,N
        E=FLOAT(J)
        EE=E*E
        EEE=E*EE
        F(J)=FB(J)
        CT(1,J)=1.0
        CT(2,J)=E
        CT(3,J)=EE
    8   CT(4,J)=EEE
C EXAMPLE 4.
      GO TO 80
    9 N=28
      M=3
      M=4
      M=2
      DO 10 J=1,N
        E=FLOAT(J)
        EE=E*E
        EEE=E*EE
        F(J)=FC(J)
        CT(1,J)=1.0
        CT(2,J)=E
        CT(3,J)=EE
   10   CT(4,J)=EEE
   80 CONTINUE
      M1=M+1
      WRITE(3,122)
      WRITE(3,101)IEXMPL,N,M
      WRITE(3,102)
      WRITE(3,120)(F(J),J=1,N)
      WRITE(3,103)
      DO 91 I=1,M1
   91   WRITE(3,120)(CT(I,J),J=1,N)
      NPIECE=4
C INITIAL VALUE OF VECTOR 'IXL' IS CALCULATED BY HAVING
C EQUAL OR NEAR EQUAL NUMBER OF POINTS FOR EACH SEGMENT.
      IXL(1)=1
      JJ=N/NPIECE
      DO 11 I=2,NPIECE
   11   IXL(I)=(I-1)*JJ+1
      CALL SETCLK(ORIGIN)
```

```
      CALL PC2(M,N,MM,NN,KK,CT,F,EPS,IRANK,ITER,
     1IND,CTN,FN,CTP,FP,
     2AL,BINVL,IRBASL,ICBASL,RL,IBOUNL,BVL,
     3AR,BINVR,IRBASR,ICBASR,RR,IBOUNR,BVR,
     4AP,BINVP,IRBASP,ICBASP,RP,IBOUNP,IRANKP,
     5ZP,IXL,ICHANG,V,NPIECE)
      CALL RDCLK(TIME,ORIGIN)
      WRITE(3,104)TIME
      WRITE(3,113)NPIECE
      WRITE(3,112)(IXL(J),J=1,NPIECE)
      WRITE(3,118)(ZP(I),I=1,NPIECE)
      WRITE(3,130)(IRANKP(I),I=1,NPIECE)
      WRITE(3,119)(RP(J),J=1,N)
      DO 72 I=1,NPIECE
        WRITE(3,114)I,(ICBASP(I,J),J=1,M1)
        WRITE(3,115)I,(IRBASP(I,J),J=1,M1)
 72     WRITE(3,116)I,(AP(I,J),J=1,M1)
        WRITE(3,117)
      DO 74 I=1,NPIECE
        DO 73 J=1,M1
 73       WRITE(3,120)(BINVP(I,J,K),K=1,M1)
 74     CONTINUE
      GO TO 1
100   STOP
      END
```

```
      SUBROUTINE PC2(M,N,MM,NN,KK,CT,F,EPS,IRANK,ITER,
     1IND,CTN,FN,CTP,FP,
     2AL,BINVL,IRBASL,ICBASL,RL,IBOUNL,BVL,
     3AR,BINVR,IRBASR,ICBASR,RR,IBOUNR,BVR,
     4AP,BINVP,IRBASP,ICBASP,RP,IBOUNP,IRANKP,
     5ZP,IXL,ICHANG,V,NPIECE)
C
C      GIVEN:1)AN INTEGER NUMBER 'NPIECE' WHICH IS THE NUMBER
C OF  SEGMENTS  IN  THE APPROXIMATION AND 2) INITIAL INDEXES
C OF THE FIRST POINT OF EACH OF THE  NPIECE  SEGMENTS.  THE
C SUBROUTINE  PC2  CALCULATES  A  LINEAR PIECEWISE CHEBYSHEV
C APPROXIMATION TO THE DISCRETE POINT SET {X,Y}, SUCH THAT A
C BALANCED  CHEBYSHEV ERROR NORM IS ATTEMPTED FOR THE NPIECE
C SEGMENTS (PIECES).
C      THE SUBROUTINE PC2 USES PARAMETRIC LINEAR  PROGRAMMING
C TECHNIQUES   TO  AN  EXISTING  DISCRETE  LINEAR  CHEBYSHEV
C APPROXIMATION SUBROUTINE.  IT ALSO MAKES USE OF THE  EQUI-
C OSCILLATION PROPERTY OF THE CHEBYSHEV APPROXIMATION.
C       THE  SUBROUTINE  PC2  CALCULATES  ONLY   DISCONNECTED
C PIECEWISE APPROXIMATION.
C     THE SUBROUTINE IS WRITTEN IN SINGLE PRECISION.  IT  MAY
C BE  CONVERTED  TO DOUBLE PRECISION BY MERELY REPLACING THE
C STATEMENT
C      DOUBLE PRECISION S,SA
C BY THE STATEMENT
C      IMPLICIT REAL*8(A-H,O-Z)
C      THE INPUT DATA TO THE SUBROUTINE.
C M       THE NUMBER OF TERMS IN THE APPROXIMATING FUNCTION.
C N       THE TOTAL NUMBER OF POINTS  TO  BE  PIECEWISE  APP-
C         ROXIMATED.
C MM      AN INTEGER GREATER THAN OR EQUAL TO (M+1).
C NN      AN INTEGER GREATER THAN OR EQUAL TO N.
C KK      AN INTEGER LARGE ENOUGH SUCH THAT THE TOTAL  NUMBER
C         OF SEGMENTS IS TO BE LE.KK.
C CT      A MATRIX OF DIMENSIONS MM BY NN.  ITS FIRST M  ROWS
C         AND  FIRST  N  COLUMNS  CONTAIN  THE  TRANSPOSE  OF
C         MATRIX C OF THE SYSTEM C*A=F.  MATRIX CT  DOES  NOT
C         CHANGE IN THE COMPUTATION.
C F       AN NN VECTOR.  ITS FIRST  N  ELEMENTS  CONTAIN  THE
C         R.H.S.   OF  THE  SYSTEM  C*A=F.  VECTOR F DOES NOT
C         CHANGE IN THE COMPUTATION.
C NPIECE AN INTEGER WHICH IS THE NUMBER OF SEGMENTS (PIECES)
C         OF THE APPROXIMATION;  NPIECE.LE.KK.
C IXL     A KK-VECTOR.  ON ENTRY, ITS FIRST  NPIECE  ELEMENTS
C         CONTAIN  THE  INITIAL  VALUES OF THE INDEXES OF THE
C         FIRST ELEMENTS OF THE NPIECE SEGMENTS.  SEE IXL  IN
C         THE RESULTS OF THE PROBLEM BELOW.
C EPS     A SPECIFIED TOLERANCE SUCH THAT A CALCULATED NUMBER
C         X  IS  CONSIDERED ZERO IF !X!<EPS.  FOR THE VAX 11/
C         780 THE ROUND-OFF LEVEL IS ABOUT 1.E-6  AND  1.E-16
C         FOR  SINGLE AND DOUBLE PRECISION RESPECTIVELY.  FOR
```

```
C         THIS MACHINE, EPS IS USUALLY TAKEN 1.E-4 AND 1.E-11
C         FOR SINGLE AND DOUBLE PRECISION CALCULATION RESPCT.
C      THE RESULTS OF THE PROBLEM.
C IXL     A KK-VECTOR. ON EXIT, ITS FIRST NPIECE ELEMENTS
C         CONTAIN THE FINAL VALUES OF THE INDEXES OF THE
C         FIRST ELEMENTS OF THE NPIECE SEGMENTS. FOR EXAMPLE
C         IF IXL=(1,5,12,22, ... ), THEN THE FIRST SEGMENT
C         CONTAINS POINTS 1 TO 4, THE SECOND SEGMENT CONTAINS
C         THE POINTS 5 TO 11, THE THIRD SEGMENT CONTAINS THE
C         POINTS 12 TO 21 ... .
C IRANKP AN KK VECTOR. ITS FIRST NPIECE ELEMENTS CONTAIN
C         THE RANK VALUES OF MATRICES C IN THE EQUATIONS
C         C*A=F FOR THE NPIECE SEGMENTS. IN CASE OF PERFECT
C         FIT FOR SEGMENT JSAY, WE GET IRANKP(J)=0.
C RP      AN NN VECTOR. ITS FIRST N ELEMENTS CONTAIN THE
C         RESIDUAL VALUES OF THE N GIVEN POINTS OF THE
C         GIVEN CURVE.
C AP      AN KK BY MM MATRIX. THE FIRST M ELEMENTS OF ITS
C         FIRST ROW CONTAIN THE ELEMENTS OF THE SOLUTION
C         VECTOR A* FOR THE FIRST SEGMENT. THE FIRST M
C         ELEMENTS OF ITS SECOND ROW CONTAIN THE SOLUTION
C         VECTOR A* FOR THE SECOND SEGMENT AND SO ON. IF THE
C         FIRST M ELEMENTS OF ROW J SAY, ARE ALL ZEROS, THIS
C         INDICATES THAT VECTOR A* OF ROW J IS NOT CALCULATED
C         BECAUSE THE NUMBER OF POINTS OF SEGMENT J IS <(M+1)
C         AND THERE IS A PERFECT FIT FOR SEGMENT J.
C ZP      AN KK VECTOR. ITS FIRST NPIECE ELEMENTS CONTAIN
C         THE NPIECE OPTIMUM VALUES OF THE CHEBYSHEV NORMS
C         FOR THE NPIECE SEGMENTS. IF ZP(J)=0.0, THIS INDI-
C         CATES THAT THERE IS A PERFECT FIT FOR SEGMENT J.
C      THE MEANING OF THE OTHER PARAMETERS.
C CTN     AN MM BY NN MATRIX. ON EXIT CTN CONTAINS THE FINAL
C         VALUES OF THE SIMPLEX TABLEAU.
C FN      AN NN VECTOR. ON EXIT FN HAS THE SAME ELEMENTS OF
C         VECTOR F ABOVE BUT WITH PLUS OR MINUS SIGNS.
C ICBAS,IRBAS,BINV,BV,IBOUND,IRANK,ITER,R,A,Z,IND:       ARE
C         PARAMETRS TO BE PASSED TO THE AUXILIARY SUBROUTINE
C         'LINFIN' AND THEY HAVE THE SAME MEANINGS AS IN THE
C         SUBROUTINE LINF. SUBROUTINE LINFIN IS A SLIGHTLY
C         MODIFIED VERSION OF SUBROUTINE LINF FOR DISCRETE
C         LINEAR CHEBYSHEV APPROXIMATION.
C V       AN MM WORKING VECTOR.
C ICBASL,IRBASL,BINVL,BVL,IRANKL,AL,ZL: HAVE THE SAME MEAN-
C         INGS RESPECTIVELY AS THE PARAMETERS ICBAS, IRBAS,
C         BINV, BV, IRANK, A, Z IN THE SUBROUTINE LINF, BUT
C         THEY ARE FOR A LEFT HAND SEGMENT.
C ICBASR,IRBASR,BINVR,BVR,IRANKR,AR,ZR: HAVE THE SAME MEAN-
C         INGS RESPECTIVELY AS THE PARAMETERS ICBAS, IRBAS,
C         BINV, BV, IRANK, A, Z IN THE SUBROUTINE LINF, BUT
C         THEY ARE FOR A RIGHT HAND SEGMENT.
C CTP     SAME AS CTN ABOVE.
```

```
C FP      SAME AS FN ABOVE.
C BINVP   AN KK BY MM BY MM ARRAY.   BINVP(1,MM,MM) CONTAINS
C         THE  FINAL  VALUES OF MATRIX BINV OF THE FIRST SEG-
C         MENT.  BINVP(2,MM,MM) CONTAINS THE FINAL VALUES  OF
C         MATRIX BINV OF SECOND SEGMENT AND SO ON.
C IRBASP AN KK BY MM MATRIX.  THE FIRST  (M+1)  ELEMENTS  OF
C         ITS  FIRST  ROW  CONTAIN  THE FINAL VALUES OF IRBAS
C         OF THE FIRST SEGMENT.   THE  FIRST  (M+1)  ELEMENTS
C         OF  ITS  SECOND  ROW  CONTAIN  THE  FINAL VALUES OF
C         IRBAS OF THE SECOND SEGMENT AND SO ON.
C ICBASP SAME  AS  IRBAS  BUT  FOR  THE  VALUES   OF   ICBAS
C IBOUNP AN  NN  VECTOR ITS FIRST N VALUES CONTAIN THE FINAL
C         VALUES  OF  THE  IBOUND  VECTORS  FOR  THE   NPIECE
C         SEGMENTS.
C ICHANG AN KK WORKING INTEGER VECTOR.
C
      DOUBLE PRECISION S,SA
      DIMENSION CT(MM,NN),F(NN)
      DIMENSION CTT(26,250),FF(250)
      DIMENSION CTN(MM,NN),CTP(MM,NN),FN(NN),FP(NN)
      DIMENSION AL(MM),BINVL(MM,MM),IRBASL(MM),ICBASL(MM),
     1RL(NN),IBOUNL(NN),BVL(MM)
      DIMENSION AR(MM),BINVR(MM,MM),IRBASR(MM),ICBASR(MM),
     1RR(NN),IBOUNR(NN),BVR(MM)
      DIMENSION IRBASP(KK,MM),ICBASP(KK,MM),BINVP(KK,MM,MM),
     1IBOUNP(NN),IRANKP(KK),RP(NN),AP(KK,MM),ZP(KK),IXL(KK),
     2ICHANG(KK),V(MM)
C
      IBIG=1000000
      M1=M+1
      KASE1=0
      KASE2=0
      DO 5 K=1,NPIECE
        ICHANG(K)=1
        IS=IXL(K)
        IF(K.EQ.NPIECE) GO TO 1
        IE=IXL(K+1)-1
        GO TO 2
    1   IE=N
    2   DO 3 J=IS,IE
          FP(J)=F(J)
          DO 3 I=1,M
    3       CTP(I,J)=CT(I,J)
      CALL LINFIN(M,N,IS,IE,MM,NN,CTP,FP,EPS,ICBASL,
     1IRBASL,BINVL,BVL,IBOUNP,IRANKL,ITER,RP,AL,ZL,IND,
     2KASE1,KASE2)
        IRANKP(K)=IRANKL
        ZP(K)=ZL
        DO 4 J=1,M1
          AP(K,J)=AL(J)
          IRBASP(K,J)=IRBASL(J)
```

```
            ICBASP(K,J)=ICBASL(J)
            DO 4 I=1,M1
    4         BINVP(K,I,J)=BINVL(I,J)
    5     CONTINUE
C BALANCING THE ERROR NORMS.
        ISTART=1
        IEND=NPIECE-1
        IPCP1=NPIECE+1
        IXL(IPCP1)=N+1
    6   DO 44 KL=ISTART,IEND,2
          KLP1=KL+1
          IF(ICHANG(KL).EQ.0.AND.ICHANG(KLP1).EQ.0) GO TO 44
          CON=ZP(KLP1)-ZP(KL)
          IF(CON.LT.0.0) CON=-CON
          IF(CON.LT.EPS) GO TO 43
          ZMAX1=ZP(KL)
          IF(ZP(KLP1).GT.ZP(KL)) ZMAX1=ZP(KLP1)
          ISL=IXL(KL)
          ISR=IXL(KLP1)
          IEL=ISR-1
          IF(KL.NE.IEND) IER=IXL(KL+2)-1
          IF(KL.EQ.IEND) IER=N
          DO 7 J=1,M1
            ICBASL(J)=ICBASP(KL,J)
            IRBASL(J)=IRBASP(KL,J)
            ICBASR(J)=ICBASP(KLP1,J)
            IRBASR(J)=IRBASP(KLP1,J)
            BVL(J)=BINVP(KL,J,M1)
            BVR(J)=BINVP(KLP1,J,M1)
            DO 7 I=1,M1
              BINVL(I,J)=BINVP(KL,I,J)
    7         BINVR(I,J)=BINVP(KLP1,I,J)
          DO 8 J=ISL,IEL
            IBOUNL(J)=IBOUNP(J)
            FN(J)=FP(J)
            DO 8 I=1,M1
    8         CTN(I,J)=CTP(I,J)
          DO 9 J=ISR,IER
            IBOUNR(J)=IBOUNP(J)
            FN(J)=FP(J)
            DO 9 I=1,M1
    9         CTN(I,J)=CTP(I,J)
          IF(ZP(KL).GT.ZP(KLP1)) GO TO 21
C Z(I)<Z(I+1)
          JMINR=IBIG
          LARGE=2
          DO 10 I=1,M1
            J=ICBASR(I)
            IF(J.EQ.0) GO TO 10
            IF(J.LT.JMINR) IMINR=I
            IF(J.LT.JMINR) JMINR=J
```

```
10      CONTINUE
      KASE2L=-1
      KASE2R=IMINR
      IF(IRANKP(KL).LT.M) KASE2L=0
      IF(IRANKP(KLP1).LT.M) KASE2R=0
      IFLAGR=0
      IFLAGL=0
      DO 14 J=ISR,JMINR
        JK=J
        IF(J.EQ.JMINR) IFLAGR=1
        IBOUNL(J)=1
        FN(J)=F(J)
        CT(M1,J)=1.0
        SA=-F(J)-ZP(KL)
        DO 11 I=1,M1
          K=IRBASL(I)
          V(I)=0.0
          IF(K.EQ.0) GO TO 11
          V(I)=CT(K,J)
          SA=SA+AP(KL,K)*CT(K,J)
11        CONTINUE
        ZJ=SA
        DO 13 I=1,M1
          S=0.0
          DO 12 K=1,M1
12          S=S+BINVL(I,K)*V(K)
13        CTN(I,J)=S
        RL(J)=ZJ
        IF(ZJ.LT.0.0) ZJ=-ZJ
        IF(ZJ.LT.ZP(KL)) GO TO 14
        IFLAGL=1
        GO TO 15
14      CONTINUE
15    IELN=JK
      ISRN=JK+1
      N1L=ISR
      N2L=IELN
      IF(IFLAGL.EQ.0) ZL=ZP(KL)
      IF(IFLAGL.EQ.0) GO TO 18
      IF(KASE2L.NE.0) GO TO 17
      DO 16 J=ISL,IELN
        FN(J)=F(J)
        DO 16 I=1,M1
16        CTN(I,J)=CT(I,J)
17    CALL LINFIN(M,N,ISL,IELN,MM,NN,CTN,FN,EPS,ICBASL,
     1IRBASL,BINVL,BVL,IBOUNL,IRANKL,ITERL,RL,AL,ZL,INDL,
     2KASE1,KASE2L)
18    IF(ZL.GT.ZMAX1) GO TO 43
      IF(IFLAGR.EQ.0) ZR=ZP(KLP1)
      IF(IFLAGR.EQ.0) GO TO 33
      NU=IER-ISRN+1
```

```
      IF(NU.EQ.M1) KASE2R=0
      IF(KASE2R.NE.0) GO TO 20
      DO 19 J=ISRN,IER
        FN(J)=F(J)
        DO 19 I=1,M1
   19     CTN(I,J)=CT(I,J)
   20 CALL LINFIN(M,N,ISRN,IER,MM,NN,CTN,FN,EPS,ICBASR,
     1IRBASR,BINVR,BVR,IBOUNR,IRANKR,ITERR,RR,AR,ZR,INDR,
     2KASE1,KASE2R)
      GO TO 33
C Z(I)>Z(I+1)
   21 JMAXL=-1
      LARGE=1
      DO 22 I=1,M1
        J=ICBASL(I)
        IF(J.EQ.0) GO TO 22
        IF(J.GT.JMAXL) IMAXL=I
        IF(J.GT.JMAXL) JMAXL=J
   22   CONTINUE
      KASE2L=IMAXL
      KASE2R=-1
      IF(IRANKP(KL).LT.M) KASE2L=0
      IF(IRANKP(KLP1).LT.M) KASE2R=0
      IFLAGL=0
      IFLAGR=0
      DO 26 L=JMAXL,IEL
        J=IEL+JMAXL-L
        IF(J.EQ.JMAXL) IFLAGL=1
        IBOUNR(J)=1
        FN(J)=F(J)
        CT(M1,J)=1.0
        SA=-F(J)-ZP(KLP1)
        DO 23 I=1,M1
          K=IRBASR(I)
          V(I)=0.0
          IF(K.EQ.0) GO TO 23
          V(I)=CT(K,J)
          SA=SA+AP(KLP1,K)*CT(K,J)
   23     CONTINUE
        ZJ=SA
        DO 25 I=1,M1
          S=0.0
          DO 24 K=1,M1
   24       S=S+BINVR(I,K)*V(K)
   25     CTN(I,J)=S
        RR(J)=ZJ
        IF(ZJ.LT.0.0) ZJ=-ZJ
        IF(ZJ.LT.ZP(KLP1)) GO TO 26
        IFLAGR=1
        GO TO 27
   26   CONTINUE
```

```
   27   ISRN=J                                                          27
        IELN=J-1
        N1R=ISRN
        N2R=IEL
        NU=IELN-ISL+1
        IF(NU.EQ.M1) KASE2L=0
        IF(IFLAGL.EQ.0) ZL=ZP(KL)
        IF(IFLAGL.EQ.0) GO TO 30
        IF(KASE2L.NE.0) GO TO 29
        DO 28 J=ISL,IELN
          FN(J)=F(J)
          DO 28 I=1,M1
   28       CTN(I,J)=CT(I,J)
   29   CALL LINFIN(M,N,ISL,IELN,MM,NN,CTN,FN,EPS,ICBASL,
       1IRBASL,BINVL,BVL,IBOUNL,IRANKL,ITERL,RL,AL,ZL,INDL,
       2KASE1,KASE2L)
   30   IF(ZL.GT.ZMAX1) GO TO 43
        IF(IFLAGR.EQ.0) ZR=ZP(KLP1)
        IF(IFLAGR.EQ.0) GO TO 33
        IF(KASE2R.NE.0) GO TO 32
        DO 31 J=ISRN,IER
          FN(J)=F(J)
          DO 31 I=1,M1
   31       CTN(I,J)=CT(I,J)
   32   CALL LINFIN(M,N,ISRN,IER,MM,NN,CTN,FN,EPS,ICBASR,
       1IRBASR,BINVR,BVR,IBOUNR,IRANKR,ITERR,RR,AR,ZR,INDR,
       2KASE1,KASE2R)
   33   IF(ZR.GT.ZMAX1) GO TO 43
        ZP(KL)=ZL
        ZP(KLP1)=ZR
        IXL(KLP1)=ISRN
        ISR=ISRN
        IEL=ISR-1
        ICHANG(KL)=1
        ICHANG(KLP1)=1
        IF(IFLAGL.EQ.0) GO TO 36
        IRANKP(KL)=IRANKL
        DO 34 J=1,M1
          ICBASP(KL,J)=ICBASL(J)
          IRBASP(KL,J)=IRBASL(J)
          AP(KL,J)=AL(J)
          DO 34 I=1,M1
   34       BINVP(KL,I,J)=BINVL(I,J)
        DO 35 J=ISL,IEL
          RP(J)=RL(J)
          IBOUNP(J)=IBOUNL(J)
          FP(J)=FN(J)
          DO 35 I=1,M1
   35       CTP(I,J)=CTN(I,J)
   36   IF(LARGE.EQ.1) GO TO 38
        DO 37 J=N1L,N2L
```

```
          IBOUNP(J)=IBOUNL(J)
          RP(J)=RL(J)
          FP(J)=FN(J)
          DO 37 I=1,M1
37          CTP(I,J)=CTN(I,J)
38      IF(IFLAGR.EQ.0) GO TO 41
        IRANKP(KLP1)=IRANKR
        DO 39 J=1,M1
          ICBASP(KLP1,J)=ICBASR(J)
          IRBASP(KLP1,J)=IRBASR(J)
          AP(KLP1,J)=AR(J)
          DO 39 I=1,M1
39          BINVP(KLP1,I,J)=BINVR(I,J)
        DO 40 J=ISR,IER
          RP(J)=RR(J)
          IBOUNP(J)=IBOUNR(J)
          FP(J)=FN(J)
          DO 40 I=1,M1
40          CTP(I,J)=CTN(I,J)
        GO TO 44
41      IF(LARGE.EQ.2) GO TO 44
        DO 42 J=N1R,N2R
          IBOUNP(J)=IBOUNR(J)
          RP(J)=RR(J)
          FP(J)=FN(J)
          DO 42 I=1,M1
42          CTP(I,J)=CTN(I,J)
        GO TO 44
43      ICHANG(KL)=0
        ICHANG(KLP1)=0
44      CONTINUE
      IS=2
      IF(IEND.EQ.1) ISTART=2
      IF(ISTART.EQ.2) IS=1
      DO 45 J=1,NPIECE
        IF(ICHANG(J).EQ.0) GO TO 45
        ISTART=IS
        GO TO 6
45  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LINFIN(M,N,N1,N2,MM,NN,CT,F,EPS,ICBAS,
     1IRBAS,BINV,BV,IBOUND,IRANK,ITER,R,A,Z,IND,
     2KASE1,KASE2)
C     THIS SUBROUTINE IS A SLIGHT MODIFICATION TO THE
C SUBROUTINE LINF (SEE TEXT). IT SOLVES AN OVERDETERMINED
C SYSTEM OF LINEAR EQUATIONS IN THE MINMAX NORM, CHEBYSHEV
C NORM.  IT USES A MODIFIED SIMPLEX METHOD TO THE LINEAR
C PROGRAMMING FORMULATION OF THE PROBLEM.
C     THE SYSTEM OF LINEAR EQUATIONS HAS THE FORM
C                          C*A=F,
C WHERE C IS A GIVEN REAL N BY M MATRIX  OF  RANK  =(OR<)M<N
C AND F IS A GIVEN REAL N-VECTOR.
C     THE PROBLEM  IS  TO  CALCULATE  THE  ELEMENTS  OF  THE
C M VECTOR A WHICH GIVES THE MINIMUM NORM Z.
C              Z=MAX!R(I)!, I=1,2,...,N,
C WHERE R(I) IS THE ITH RESIDUAL AND IS GIVEN BY
C   R(I)=C(I,1)*A(1)+C(I,2)*A(2)+ ...  +C(I,M)*A(M)-F(I).
C     THE INPUT DATA TO THE SUBROUTINE.
C M       THE NUMBER OF COLUMNS OF MATRIX IN SYSTEM C*A=F.
C N       THE NUMBER OF ROWS OF MATRIX IN SYSTEM C*A=F.
C N1      AN INTEGER WHICH IS THE INDEX OF  THE  FIRST  POINT
C         OF THE SEGMENT AT HAND.
C N2      AN INTEGER WHICH IS THE  INDEX  OF  THE  END  POINT
C         OF THE SEGMENT AT HAND.
C MM      AN INTEGER GREATER THAN OR EQUAL TO (M+1).
C NN      AN INTEGER GREATER THAN OR EQUAL TO N.
C CT      A MATRIX OF DIMENSIONS MM BY  NN.  ON  ENTRY,  ITS
C         FIRST  M  ROWS  AND  FIRST  N  COLUMNS  CONTAIN THE
C         TRANSPOSE OF MATRIX C OF THE SYSTEM C*A=F.
C F       AN NN-VECTOR.  ON  ENTRY,  ITS  FIRST  N  ELEMENTS
C         CONTAIN THE R.H.S.  OF THE SYSTEM C*A=F.
C EPS     A SPECIFIED TOLERANCE SUCH THAT A CALCULATED NUMBER
C         X  IS  CONSIDERED ZERO IF !X!<EPS.  FOR THE VAX 11/
C         780 THE ROUND-OFF LEVEL IS ABOUT 1.E-6  AND  1.E-16
C         FOR  SINGLE AND DOUBLE PRECISION RESPECTIVELY.  FOR
C         THIS MACHINE, EPS IS USUALLY TAKEN 1.E-4 AND 1.E-11
C         FOR SINGLE AND DOUBLE PRECISION CALCULATION RESPCT.
C KASE1, KASE2 ARE TWO INTEGERS.  SEE COMMENTS AFTER  STATE-
C         MENT ITEST=1 BELOW.
C     THE RESULTS OF THE PROBLEM.
C IRANK   THE CALCULATED RANK OF MATRIX C.
C ITER    THE NO.  OF ITERATIONS, OR THE NO.  OF  TIMES  THE
C         SIMPLEX  TABLEAU IS CHANGED BY A GAUSS-JORDAN STEP.
C A       AN MM-VECTOR.  ON EXIT, THE FIRST M ELEMENTS  OF  A
C         CONSTITUTE  THE SOLUTION TO THE GIVEN SYSTEM C*A=F.
C R       AN NN-VECTOR.  THE FIRST N ELEMENTS OF R CONSTITUTE
C         THE CHEBYSHEV RESIDUAL VECTOR (C*A-F).
C Z       THE MINIMUM CHEBYSHEV NORM OF THE  RESIDUALS  R(I).
C IND     A  RETURN  INDICATOR.  IND=0  INDICATES  THAT  THE
C         SOLUTION VECTOR A IS UNIQUE.  IND=1, INDICATES THAT
C         THE  SOLUTION VECTOR A IS MOST PROBABLY NOT UNIQUE.
```

```
C         IND=-1 INDICATES THAT NO FEASIBLE SOLUTION HAS BEEN
C         FOUND DUE TO POSSIBLE ILL-CONDITIONING OF MATRIX C.
C      THE MEANING OF SOME OF THE AUXILIARY PARAMETERS.
C BINV    AN MM SQUARE MATRIX, ITS FIRST (M+1) ROWS AND FIRST
C         (M+1)  COLUMNS  CONTAIN  THE  INVERSE  OF THE BASIS
C         MATRIX IN THE LINEAR PROGRAMMING PROBLEM.
C BV      AN MM-VECTOR.  ITS FIRST (M+1) ELEMENTS CONTAIN THE
C         BASIC  SOLUTION  IN THE LINEAR PROGRAMMING PROBLEM.
C ICBAS   AN MM-VECTOR.  ITS FIRST (M+1) ELEMENTS CONTAIN THE
C         INDICES OF THE COLUMNS OF CT FORMING THE BASIS.
C     THE SUBROUTINE IS WRITTEN IN SINGLE PRECISION.  IT  MAY
C BE  CONVERTED  TO DOUBLE PRECISION BY MERELY REPLACING THE
C STATEMENT
C      DOUBLE PRECISION S
C BY THE STATEMENT
C      IMPLICIT REAL*8(A-H,O-Z)
       DOUBLE PRECISION S
       DIMENSION CT(MM,NN),F(NN),R(NN),A(MM),BINV(MM,MM)
      *,BV(MM),IBOUND(NN),ICBAS(MM),IRBAS(MM)
       BIGNUM=1./(EPS*EPS)
       ITER=0
       IPART=1
       KL=1
       IRANK=M
       M1=M+1
       IBV=0
       IND=0
       ITEST=1
C      KASE1.EQ.0.AND.KASE2.EQ.0 INDICATES THAT THE  ORIGINAL
C SUBROUTINE LINF IS USED.
C      KASE1.NE.0.AND KASE2.EQ.0 INDICATES THAT  THE  SEGMENT
C AT  HAND  IS  ENLARGING  BY  ADDING  POINTS TO IT FROM THE
C RIGHT END
C      KASE1.EQ.0.AND KASE2.LT.0 INDICATES THAT  THE  SEGMENT
C AT  HAND  IS  ENLARGING  BY  ADDING  POINTS TO IT FROM THE
C RIGHT END OR FROM THE LEFT END.  THIS DEPENDS ON  THE  NEW
C VALUES OF THE PARAMETERS N1 AND N2 OF THE SEGMENT.
C      KASE1.EQ.0.AND.KASE2.GT.0 INDICATES THAT  THE  SEGMENT
C AT  HAND  IS  SHRINKING  BY  DELETING POINTS FROM ITS LEFT
C END OR FROM ITS RIGHT END.  THIS DEPENDS ON THE NEW VALUES
C OF THE PARAMETERS N1 AND N2.
C      MAKE SURE IN THE CALLING STATEMENT  THAT  IBOUND(J)=1,
C WHERE  J  ARE  THE  INDEXES  OF  THE  ADDED  POINTS TO THE
C SEGMENT.
       IF(KASE1.NE.0.OR.KASE2.NE.0) ITER=M1
       IF(KASE1.NE.0) GO TO 19
       IF(KASE2.LT.0) GO TO 19
       IF(KASE2.GT.0) IOUT=KASE2-1
       IF(KASE2.GT.0) GO TO 5
C
       DO 1 I=1,M1
```

```
      A(I)=0.
    1 IRBAS(I)=I
      DO 3 J=1,M1
        DO 2 I=1,M1
    2     BINV(I,J)=0.
    3   BINV(J,J)=1.
      DO 4 J=N1,N2
        CT(M1,J)=1.
    4   IBOUND(J)=1
      IOUT=0
    5 IOUT=IOUT+1
      IF(IOUT.GT.M1) GO TO 15
      PIV=0.
      DO 6 J=N1,N2
        D=CT(IOUT,J)
        IF(D.LT.0.0) D=-D
        IF(D.LE.PIV) GO TO 6
        JIN=J
        PIV=D
    6   CONTINUE
C DETECTION OF RANK DEFICIENCY.
      IF(PIV.GT.EPS) GO TO 36
      IF(IOUT.EQ.M1) GO TO 10
      DO 7 J=N1,N2
        D=CT(IOUT,J)
        CT(IOUT,J)=CT(KL,J)
    7   CT(KL,J)=D
      K=IRBAS(IOUT)
      IRBAS(IOUT)=IRBAS(KL)
      IRBAS(KL)=0
      DO 8 J=KL,M
        BINV(IOUT,J)=BINV(KL,J)
    8   BINV(KL,J)=0.
      ICBAS(IOUT)=ICBAS(KL)
      ICBAS(KL)=0
      DO 9 I=KL,M1
        BINV(I,IOUT)=BINV(I,KL)
    9   BINV(I,KL)=0.
      IRANK=IRANK-1
      KL=KL+1
      GO TO 5
   10 DO 12 J=N1,N2
        DO 11 I=KL,M
          IF(J.EQ.ICBAS(I)) GO TO 12
   11     CONTINUE
        JIN=J
        GO TO 13
   12   CONTINUE
   13 F(JIN)=-F(JIN)
      IBOUND(JIN)=-IBOUND(JIN)
      DO 14 I=KL,M
```

```
   14    CT(I,JIN)=-CT(I,JIN)
      CT(M1,JIN)=2.-CT(M1,JIN)
      GO TO 36
   15 K1=KL
C PART 2 OF THE ALGORITHM.
   16 DO 18 I=K1,M1
         IF(BV(I).GT.-EPS) GO TO 18
         IBV=IBV+1
         IOUT=I
         JIN=ICBAS(I)
         F(JIN)=-F(JIN)
         IBOUND(JIN)=-IBOUND(JIN)
         DO 17 L=KL,M1
           D=BV(L)
   17      CT(L,JIN)=D+D-CT(L,JIN)
         GO TO 36
   18    CONTINUE
C PART 3 OF THE ALGORITHM.
   19 IPART=3
      DO 23 J=N1,N2
        DO 20 I=KL,M1
          IF(J.EQ.ICBAS(I)) GO TO 22
   20     CONTINUE
        S=-F(J)
        DO 21 I=KL,M1
          K=ICBAS(I)
   21     S=S+CT(I,J)*F(K)
        R(J)=S
        GO TO 23
   22   R(J)=0.
   23   CONTINUE
      S=0.
      DO 24 I=KL,M1
        K=ICBAS(I)
   24   S=S+BV(I)*F(K)
      Z=S
      IF(Z.GT.-EPS) GO TO 28
      DO 25 J=N1,N2
        F(J)=-F(J)
        IBOUND(J)=-IBOUND(J)
   25   R(J)=-R(J)
      Z=-Z
      DO 27 J=KL,M
        DO 26 I=KL,M1
   26     BINV(I,J)=-BINV(I,J)
   27   CONTINUE
C DETERMINE THE VECTOR WHICH ENTERS THE BASIS.
   28 IVO=0
      G=BIGNUM
      TZ=Z+Z+EPS
      DO 32 J=N1,N2
```

```
      DO 29 I=KL,M1
        IF(J.EQ.ICBAS(I)) GO TO 32
29      CONTINUE
      D=R(J)
      IF(D.LT.-EPS) GO TO 30
      IF(D.LT.TZ) GO TO 32
      E=TZ-D
      IF(E.GE.G) GO TO 32
      IVO=-1
      GO TO 31
30    E=D
      IF(E.GE.G) GO TO 32
      IVO=1
31    G=E
      JIN=J
32    CONTINUE
    IF(IVO.EQ.0) GO TO 45
    IF(IVO.EQ.1) GO TO 34
    DO 33 I=KL,M1
      D=BV(I)
33    CT(I,JIN)=D+D-CT(I,JIN)
    R(JIN)=Z+Z-R(JIN)
    F(JIN)=-F(JIN)
    IBOUND(JIN)=-IBOUND(JIN)
34  ITEST=0
C DETERMINE THE VECTOR WHICH LEAVES THE BASIS.
    THMAX=BIGNUM
    DO 35 I=KL,M1
      D=CT(I,JIN)
      IF(D.LT.EPS) GO TO 35
      G=BV(I)/D
      IF(G.GT.THMAX) GO TO 35
      THMAX=G
      IOUT=I
      ITEST=1
35    CONTINUE
    IF(ITEST.EQ.1) GO TO 36
    IND=-1
    RETURN
C A GAUSS-JORDAN ELIMINATION STEP.
36  PIVOT=CT(IOUT,JIN)
    DO 37 J=N1,N2
37    CT(IOUT,J)=CT(IOUT,J)/PIVOT
    L=M1
    K1=KL+ITER
    IF(K1.LT.M1) L=K1
    DO 38 J=KL,L
38    BINV(IOUT,J)=BINV(IOUT,J)/PIVOT
    DO 41 I=KL,M1
      IF(I.EQ.IOUT) GO TO 41
      D=CT(I,JIN)
```

```
      IF(D.EQ.0.0) GO TO 41
      DO 39 J=N1,N2
   39   CT(I,J)=CT(I,J)-D*CT(IOUT,J)
      DO 40 J=KL,L
   40   BINV(I,J)=BINV(I,J)-D*BINV(IOUT,J)
   41 CONTINUE
      DO 42 I=KL,M1
   42 BV(I)=BINV(I,M1)
      ITER=ITER+1
      ICBAS(IOUT)=JIN
      IF(IPART.EQ.3) GO TO 43
      IF(IPART.EQ.1.AND.IBV.EQ.0) GO TO 5
      IF(IOUT.EQ.M1) GO TO 19
      K1=IOUT+1
      GO TO 16
   43 D=R(JIN)
      DO 44 J=N1,N2
   44 R(J)=R(J)-D*CT(IOUT,J)
      Z=Z-D*BV(IOUT)
      GO TO 28
C CALCULATING THE ANSWER OF THE PROBLEM.
   45 DO 47 J=KL,M1
        S=0.
        DO 46 I=KL,M1
          K=ICBAS(I)
   46     S=S+F(K)*BINV(I,J)
        K=IRBAS(J)
   47   A(K)=S
      DO 48 J=N1,N2
        D=R(J)-Z
        IF(IBOUND(J).EQ.-1) D=-D
   48   R(J)=D
      IF(IRANK.LT.M) GO TO 50
      DO 49 I=1,M1
        IF(BV(I).LT.EPS) GO TO 50
   49   CONTINUE
      RETURN
   50 IND=1
      RETURN
      END
```

POOR

QUALITY

ORIGINAL